The Ornstein-Uhlenbeck Process In Neural Decision-Making:

Mathematical Foundations
And
Simulations Suggesting The Adaptiveness Of Robustly Integrating Stochastic Neural Evidence

Michael Thomas Wojnowicz

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science

University of Washington
2012

Committee:
Professor Eric Shea-Brown
Professor Krzysztof Burdzy

Program Authorized to Offer Degree:
Mathematics

Contents

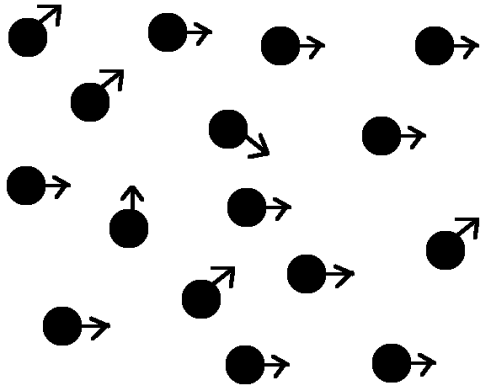
Part I

Introduction

1 Introduction

1.1 Thesis Overview

A recently hot research topic has been the neural basis of decision making (e.g. Gold and Shadlen 2007). In particular, much research has been done on a perceptual decision-making task known as the ”dot coherence” task (see Figure below.) In this task, which dots randomly move on the screen, and the animal is to determine the direction of coherent motion. Typically, the animal communicates its decision (e.g. ”leftward” or ”rightward”) by saccading (that is, moving its eyes) to a particular choice target.



How does the animal make such a decision? There appears to be neural circuit in which momentary evidence is integrated over time. The evidence used to form the direction decision is stored in the middle temporal area (MT) tuned for the direction of visual motion. Indeed, signal detection theory analyses of the strength and variability of MT responses predict behavioral accuracy and timing (Britten et al. 1992, 1993; Shadlen et al. 1996). In contrast, the lateral inteparietal area (LIP) appears to integrate the momentary directional evidence stored in MT (Gold and Shadlen 2007). When neurons in the LIP reach a threshold, the animal commits to one of the choice targets.

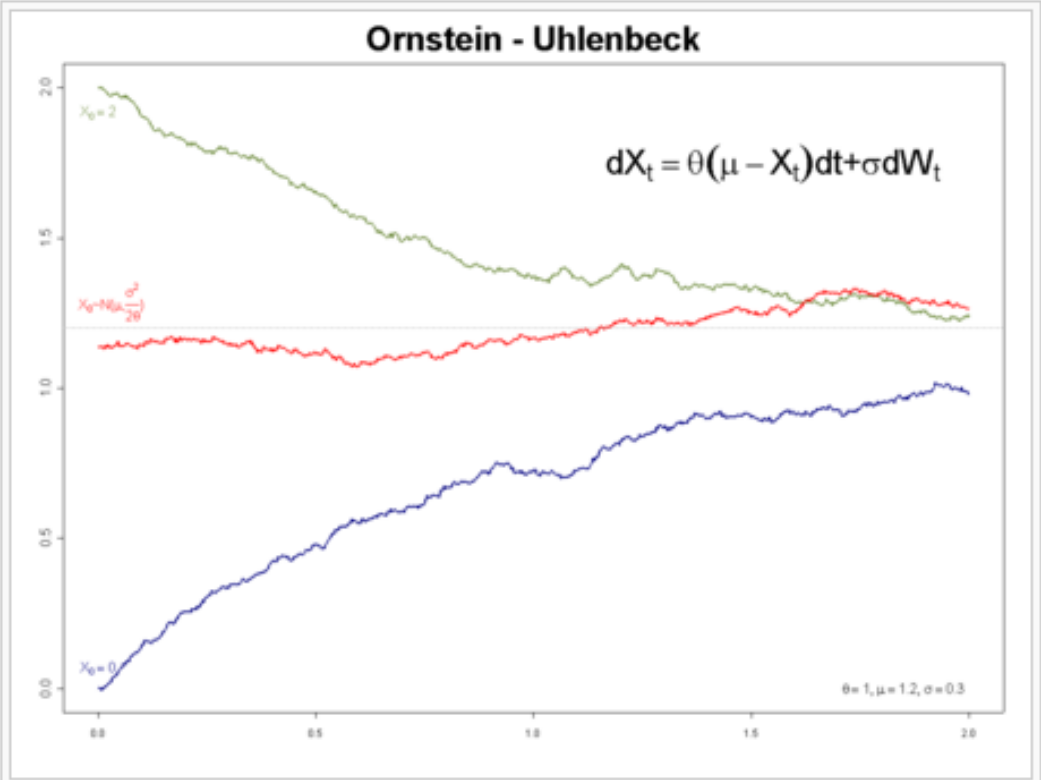
Because of the noise inherent in the stimulus, as well as in the neural representations of the stimulus, mathematical models of this decision-making process have begun to rely upon stochastic processes. The neural representation of a stimulus property is a stochastic variable which continuously evolves over time – thus, its evolution requires a model of both dynamics and probabilistic (stochastic elements). In particular, mathematical models of an evolving neural decision often utilize an Ornstein-Uhlenbeck stochastic process, x_t, which satisfies the following stochastic differential equation:

$$dx_t = \theta(\mu - x_t)\,dt + \sigma\,dW_t \tag{1.1}$$

where $\theta > 0$, μ, and $\sigma > 0$ are parameters, x_0 is the initial condition, and W_t denotes the standard Wiener process or standard Brownian motion. Several realizations of this process are shown in Figure 1.

The OU process has been used as a model for the momentary neural evidence supporting a certain decision that is fed into a neural integrator (Cain et al., 2011).

This master’s thesis aims to accomplish four goals: (1) first, we lay down the conceptual foundations of an Ornstein-Uhlenbeck process (at the level accessible to an advanced undergraduate mathematician); (2) second, we describe



three sample paths of different OU-processes with $\theta = 1$, $\mu = 1.2$, $\sigma = 0.3$:
blue: initial value $a = 0$ (a.s.)
green: initial value $a = 2$ (a.s.)
red: initial value normally distributed so that the process has invariant measure

Figure 1: Sample Paths of an OU Stochastic Process (from Wikipedia)

a particular model used to describe the stochastic accumulation of neural evidence in perceptual decision making-tasks (Cain et al., in press); (3) third, we describe recent results in theoretical neuroscience suggesting that the brain may use a 'robust integration' mechanism in LIP for integrating evidence; and (4) fourth, we report new results suggesting that there may be an additional adaptive functional role for 'robust integration.'

1.2 Why can't we just blur the deterministic trajectories?

The master's thesis begins building up mathematical concepts (at an introductory level) related to stochastic processes and stochastic differential equations. One basic and excellent question is – what's the point of all this? Can't we just understand the behavior of the neural circuit as slightly messier versions of ordinary differential equations and numerical simulations? As it turns out, this is a mistake.

> It is often thought that the action of noise merely amounts to a blurring of trajectories of the deterministic system. That is indeed the case for "observational" or "measurement" noise. However, in nonlinear systems where noise acts as a driving force, noise can drastically modify the deterministic dynamics (Longtin 2010).

Let us give an extremely simple example illustrating why can't we conceptualize stochastic differential equations as mere blurrings of their deterministic counterparts. For this example, let's imagine neurons whose outputs are exponential functions of their input (such as STN neurons in the basal ganglia; Bogacz & Gurney 2007). Now imagine that the input which these STN neurons receive is growing linearly in time, but subject to some additive white noise. So the input to the STN would be $I(t) = t + W(t)$. Since the STN transforms its input exponentially, the output from the STN would be $O(t) = e^{I(t)} = e^{t+W(t)}$.

Now what would the average sample path (that is, the expected sample path) look like? An intuitive theory that "noise blurs deterministic trajectories" would predict that the noise would cancel out, and the mean sample path would be $\mu_{\text{intuitive}}(t) = e^t$ However, using numerical methods for simulating stochastic processes, we can draw a number of sample paths, as shown in the figure below (from Higham, 2001). An approximation to the mean solution (the mean of 1,000 discretized paths) is given in blue.

As shown in the figure, the value of $O(t)$ at $t = 1$ is greater than e, which is the value expected naively. Indeed, the exact solution to the mean trajectory of the stochastic process $O(t) = e^{t+W(t)}$ can be shown (see Higham, 2001) to be

$$\mu_{\text{correct}}(t) = e^{9t/8}$$

What is the explanation? Upon a moment's reflection, it makes sense that the true mean trajectory should be above $\mu_{\text{intuitive}}$, because the white noise process W(t) is continuously increasing the value of $O(t)$ – even when W(t) takes on an enormously negative value, O(t) will be positively incremented.

So, even in this simple example with no dynamics, the noise here does not "cancel out" when we consider the average trajectory. Apparently, when we have a composition of functions – or a causal chain, as happens frequently in the brain which is continuously cascading information from one brain region to the next (McClelland 1979; O'Reilly 1998) – we will need some deeper theory to understand the effect of noise in neural circuits. This is because, as our simple example shows, noise can dramatically modify even the *mean* behavior of nonlinear systems.

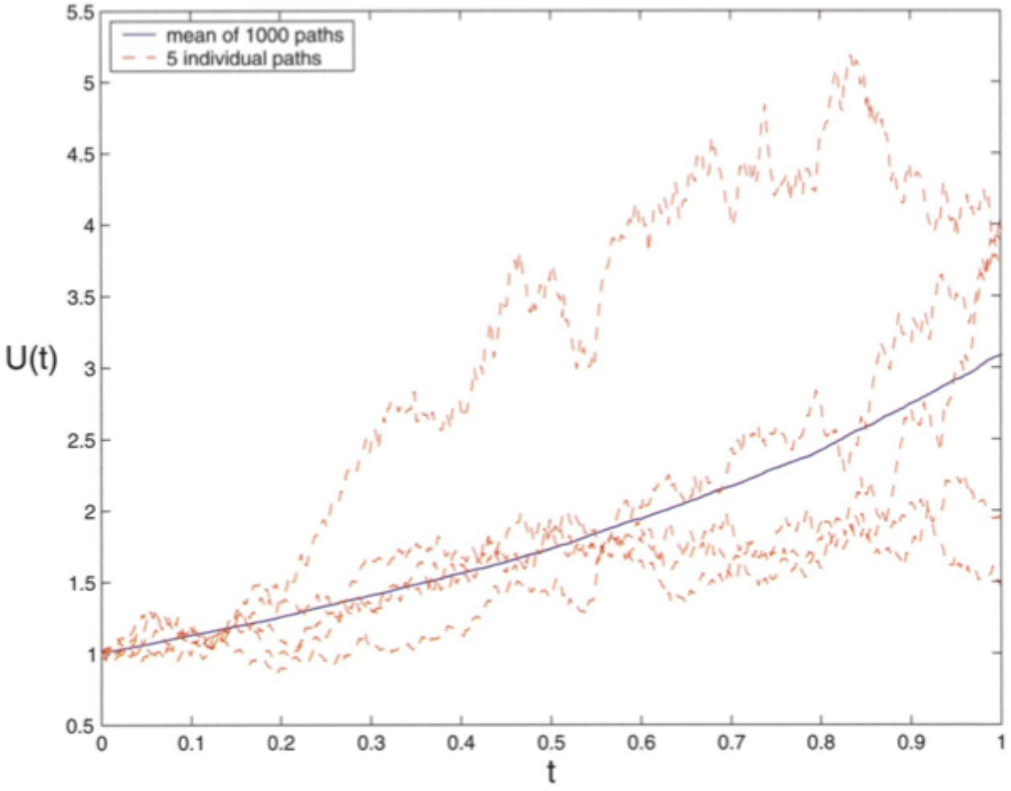


Fig. 2 *The function $u(W(t))$ averaged over* 1000 *discretized Brownian paths and along* 5 *individual paths, from* `bpath3.m`.

Figure 2: Sample Paths and Mean Trajectory of A Noisy Stochastic Process from Higham (2001)

Part II

Introductory Mathematical Foundations

2 Continuous Markov Stochastic Processes

A stochastic process is like a differential equation, but one which moves through its state space stochastically. In this section, we define stochastic processes, as well as a special class of stochastic processes called Markov stochastic processes.

2.1 What is a Stochastic Process?

Imagine an experiment with an uncertain outcome – for example, you might roll a pair of die. We might record the sum of the numbers on the die, calling this X. We might record the difference of the numbers on the die, calling this Y. In either case, we have a "random variable" – a function of an outcome of an inherently uncertain experiment. As evidenced by the random variables X and Y, any experiment may have many random variables associated with it. Technically speaking, a random variable X is function from a probability space $(\Omega, \mathscr{F}, P)$ typically to the real numbers which is measurable. For our purposes, let us focus on the "probability space." A probability space consists

of three parts: (1) a sample space Ω which is the set of all possible outcomes, (2) a set of events $\mathscr{F}$ where each event is a set containing zero or more outcomes, and (3) a probability measure P assigning a probability level to each event $F \in \mathscr{F}$.

Now imagine an experiment with three trials – for example, you might roll a pair of die three times. We might record the three outcomes as X_1, X_2, X_3, where X_i gives the sum of the numbers on the die for trial i. Whereas this gives a finite collection of consecutively recorded random variables, we could generalize these collections into an index set with larger cardinalities. In general, a *stochastic process* is a collection of random variables $(X_t(\omega), t \in T, \omega \in \Omega)$ defined on some index set T. We usually think of the index set as representing "time." If $T = \mathbb{N}$ then then the collection X_t is called a discrete-time stochastic process, and if $T = [a,b] \subset \mathbb{R}$, then the collection X_t is called a continuous-time stochastic process. We will focus on continuous stochastic processes where the index set T represents time, and therefore the collection of random variables X_t represent how the state of a stochastic variable evolves over continuous time. The state X_i may be vector valued, in which case the stochastic process represents how the state of a stochastic *system* evolves over continuous time. The use of the term system refers to the fact that the values of the full collection of state variables may interact to determine the probability distributions of any single state variable at some other time in the future or past.

Thus, stochastic processes are collections of random variables. Whereas a random variable X maps outcomes $\omega \in \Omega$ into real-valued numbers $X(\omega) \in \mathbb{R}^n$, a stochastic process maps outcomes $\omega \in \Omega$ into trajectories that are functions of time, $X_t(\omega) \in$ function space. We might imagine pulling some outcome ω out of a hat containing the set of all possible outcomes Ω, and once the outcome has been selected, it predetermines a certain "trajectory" or "sample path" throughout all time in T. Thus, technically speaking, a stochastic process is a mapping $\Omega X T \rightarrow \mathbb{R}^n$ – that is, a mapping from the set of all possible outcomes crossed with time into Euclidean Space $\mathbb{R}^n$. Thus, a stochastic process is a function of two variables, and can be considered from two different perspectives:

- *As a function of ω (fixing t)*: In this interpretation, $X_t(\omega)$ is just a random variable. The outcome has not yet been determined. But the time is fixed – we only care to observe the process at a single time, so we focus on one moment and ignore the rest of the trajectory. Once the outcome is determined, the process is just some point in the range $\mathbb{R}^n$, and assumes different values with different probabilities.
- *As a function of t (fixing ω)*: In this interpretation, $X_\omega(t)$ is an ordinary deterministic function of time. The outcome has already been determined. But time is not fixed – so observed a particular trajectory or sample path which unfolds out over time.

In summary, then we imagine a stochastic process this way: there are many possible functions of time in a hat. When an outcome ω is selected, the stochastic process tells us which function of time has been selected. This is clearly a generalization of a random variable, which we imagine in this way: there are many possible numbers in a hat. When an outcome ω is selected, the random variable tells us which number (e.g. sum of numbers on the faces of die) has been selected.

2.2 What is a Markov Stochastic Process?

The law of total probability tells us how to collapse a probability function of many random variables across all possible values of a random variable (in this case, B): $\sum_B P(A \cap B \cap C) = P(A \cap C)$. Now imagine that we have a stochastic process – a system which evolves probabilistically in time. Thus, we interpret the RV's $\{$(A);(B);(C)$\}$ as RV's evolving in time $\{(X_1,T_1);(X_2,T_2);(X_3,T_3)\}$ etc. In addition, we consider the law of total probability to now involve a continuum of values – thus, rather than discretely summing, we must integrate the random variable. Thus, for a general stochastic process measured at three time points, we would have $p(x_1,t_1|x_3,t_3) =$

$\int dx_2\, p(x_1,t_1|x_2,t_2;x_3,t_3)\, p(x_2,t_2|x_3,t_3)$. However, we might be able to simplify this equation depending upon the degree of dependency between the future and the past. Thus, we can construct a *tower of dependencies*, where D_i indicates that the future steps depends upon i steps stretching backwards in time. Because we are dealing with stochastic processes, we interpret these dependencies with respect to time:

D_∞	$p(x,t_1\|x_n,t_n) = \int dx_2\, p(x_1,t_1\|x_2,t_2;x_2,t_3;\cdots x_n,t_n)\, p(x_2,t_2\|x_3,t_3\cdots x_n,t_n)\, p(x_{n-1},t_{n-1}\|x_n,t_n).$ Complete Dependence: entire history matters
D_2 (D_i)	$p(x,t_1\|x_3,t_3) = \int dx_2\, p(x_1,t_1\|x_2,t_2;x_3,t_3)\, p(x_2,t_2\|x_3,t_3)$ Intermediate Dependence: current state plus (generally) i-1 previous states matter
D_1	$p(x,t_1\|x_3,t_3) = \int dx_2\, p(x_1,t_1\|x_2,t_2)\, p(x_2,t_2\|x_3,t_3)$ Markov Dependence: only the current state matters
D_0	$p(x,t_1\|x_3,t_3) = \int dx_2\, p(x_1,t_1)\, p(x_2,t_2). = p(x_1,t_1)$ Independence: future is independent of past and current state

When we say that something "matters," we means more specificially that it influences the probability distribution of the next upcoming state. For this project, we assume that our probability distributions are Markov (i.e. D_1). That is, the current state matters for determining the probability distribution of the next upcoming future state, but how the system reached that current state is irrelevant to the dynamics.

2.3 Moving Pollen Grains: Einstein's Example of a Markov Stochastic Process

For example, Einstein made a Markov assumption in his derivation Brownian Motion. Einstein wanted to model the movement of small pollen grains when suspended in water. He derived a frequency law expressed as follows.

$$\underbrace{f(x,t+\tau)}_{\text{(density of number of particles per unit volume)}} \underbrace{dx}_{\text{(small thickness)}} =$$

$$dx \int_{-\infty}^{\infty} \underbrace{f(x+\Delta,t)}_{\text{(density of particles at preshifted location)}} \underbrace{\phi(\Delta)}_{\text{(probability of making such a shift)}} \underbrace{d\Delta}_{\text{(small shift)}}$$

2.4 The Dynamics of Continuous Markov Stochastic Processes

Can we come up with a differential equation that governing the dynamic evolution of a general Markov stochastic process? In particular, how would the probability distribution defined over state space vary in time? In this section, we attempt to answer these questions.

Let us consider $p(z,t|y,t')$, the transitional probability of a stochastic system moving from some previous state y at some previous time t' into some focal state z at the current time t. We might want to ask how this transition

probability distribution is evolving over time. That is, we might want to analyze the dynamics of the probability distribution. This means we'd like to analyze $\partial_t\, p(z,t|y,t')$. The Differential Chapman-Kolmogorov Equation (DCKE) describes the temporal dynamics of a continuous stochastic process, so long as the process is Markov. In particular, the DCKE says:

$$\begin{aligned}\partial_t\, p(z,t|y,t') = &-\sum_i \frac{\partial}{\partial z_i}[A_i(z_i,t)p(z,t|y,t')] \\ &+\sum_{i,j}\frac{1}{2}\frac{\partial^2}{\partial z_i \partial x_j}[B_{ij}(z,t)p(z,t|y,t')] \\ &+\int dx\,[W(z|x,t)\,p(x,t|y,t') - W(x|z,t)\,p(z,t|y,t')]\end{aligned} \tag{2.1}$$

Below we interpret the constituent terms on the RHS of the equation.

2.5 The Three Drivers of Markov Dynamics: Drifts, Diffusions, and Jumps

The DCKE describes the dynamics of a Markov stochastic process. As we see in (2.1), the right hand side reveals three separate terms additively contributing to the evolution of the transitional probability distribution. These terms describe drifts, diffusions, and jumps. Below we interpret these three kinds of terms in the context of the DCKE.

2.5.1 Drift and Diffusion Coefficients: Probabilistic Steps in Taylor Series Expansions

The derivation of the DCKE equation (2.1) is instructive for interpreting the coefficients A_i and B_ij, which represent drift and diffusion, respectively. In particular, we explain how those coefficients are determined.

A stochastic process $\mathbf{x}_t$ travels through its state space in a way determined by its time-dependent probability distribution function. We imagine that this stochastic process is serves as the domain for some function $f(\mathbf{x}_t)$, which would be a "stochastic function" – a function of an uncertain input – technically, the function of a random variable. Assuming the function is well-behaved (that is, locally the function's growth can be understood through lines and parabolic arcs) , we could analyze the evolution of a stochastic function through a second-order Taylor approximation. In particular, we could analyze the evolution of the expected value of the stochastic function. But in that case, we no longer want to consider deterministic "steps" along the axes (h_i along the i-th axis, h_j along the jth axis), but *expected values* of "steps" along these axes. Thus, the drift coefficient A_i represents the expected value of a step along the i-th axis, and the diffusion coefficient B_{ij} represents the expected value of the product of steps along the i-th and j-th axes. Since the drift coefficient in the DCKE describes the mean of the probability distribution and the diffusion coefficient describes its variance, we see an intimate connection between the terms of a "probabilistic Taylor series expansion" and the moments of a probability distribution. In particular, it seems that, at least when we are analyzing the temporal evolution of a probability distribution, the n-th moment of the probability distribution comes from the n-th order term in the Taylor Series Expansion.

Let us investigate this observation more carefully. While the DCKE (2.1) describes the temporal evolution of a probability distribution, it is in fact deduced from a more general equation for the temporal evolution of the expected value of a function of a stochastic process (i.e. the random variables represent how the "state" of some system is evolving over time). We use the notation that the state of the system at time t is given by $\mathbf{x}_t$, and thus $f(\mathbf{x}_t)$ gives the

value of our function at a particular time. Thus we want to explore

$$\partial_t \int d\mathbf{x}\, f(\mathbf{x})\, p(\mathbf{x},t|\mathbf{y},t') \tag{2.1}$$

This partial derivative expression describes the temporal evolution of the expected value of a function of a random variable. By the definition of partial derivative, we get:

$$\partial_t \int d\mathbf{x}\, f(\mathbf{x})\, p(\mathbf{x},t|\mathbf{y},t') = \lim_{t\to 0} \frac{1}{\Delta t} \int d\mathbf{x}\, f(\mathbf{x})\, [p(\mathbf{x},t+\Delta t|\mathbf{y},t') - p(\mathbf{x},t|\mathbf{y},t')]$$

Yet by the assumption that our stochastic process is Markov, we know that

$$p(\mathbf{x},t+\Delta t|\mathbf{y},t') = \int d\mathbf{z}\, p(\mathbf{x},t+\Delta t|\mathbf{z},t)\, p(\mathbf{z},t|\mathbf{y},t')$$

Moreover, in general any "well-behaved" function (i.e. twice continuously differentiable function) can be expressed through a second-order Taylor series expansion. Thus we can rewrite f(x) as:

$$\boxed{f(\mathbf{x}) = f(\mathbf{z}) + \sum_i \frac{\partial f(\mathbf{z})}{\partial z_i}(h_i) + \sum_{i,j} \frac{\partial^2 f(\mathbf{z})}{\partial z_i \partial z_j}(h_i)(h_j) + |\mathbf{x}-\mathbf{z}|^2 R(\mathbf{x},\mathbf{z})}$$

By these two substitutions, our original expression (2.1) for the dynamic evolution of $f(\mathbf{x})$ becomes

$$\lim_{t\to 0} \frac{1}{\Delta t} \int d\mathbf{x} \int d\mathbf{z} \boxed{f(\mathbf{z}) + \sum_i \frac{\partial f(\mathbf{z})}{\partial z_i}(h_i) + \sum_{i,j} \frac{\partial^2 f(\mathbf{z})}{\partial z_i \partial z_j}(h_i)(h_j) + |\mathbf{x}-\mathbf{z}|^2 R(\mathbf{x},\mathbf{z})}\, p(\mathbf{x},t+\Delta t|\mathbf{z},t)\, p(\mathbf{z},t|\mathbf{y},t')$$

However, note that when we have a function of a stochastic process $\mathbf{x}_t$, the temporal evolution of our function evolves non-deterministically because the input to that function $\mathbf{x}_t$ travels non-deterministically through the function's domain. Thus, if we want to consider the temporal evolution of the expected value of our function, we will have to consider a *probabilistic Taylor series expansion*, where the Taylor Series Expansion (boxed expression) has probabilistic steps. Thus the standard representation of a first-order step along the i-th axis, h_i, must become replaced in the probabilistic context with:

$$\lim_{t\to 0} \frac{1}{\Delta t} \int d\mathbf{x}\, h_i\, p(\mathbf{x},t+\Delta t|\mathbf{z},t) \equiv A_i(\mathbf{z},t) + O(\varepsilon)$$

and the standard representation of a second-order product of steps along the i-th and j-th axes $h_i h_j$ becomes

$$\lim_{t\to 0} \frac{1}{\Delta t} \int d\mathbf{x}\, h_i h_j\, p(\mathbf{x},t+\Delta t|\mathbf{z},t) \equiv B_{ij}(\mathbf{z},t) + O(\varepsilon)$$

Thus, we replace the first and second-order "steps" from the Taylor's expansion and we get a new expression for the temporal evolution of the expected value of $f(\mathbf{x})$, a function of the state of the stochastic process:

$$\begin{aligned}\partial_t \int d\mathbf{x}\, f(\mathbf{x})\, p(\mathbf{x},t|\mathbf{y},t') = &\int d\mathbf{z} \left[\sum_i \frac{\partial f(\mathbf{z})}{\partial z_i} A_i(\mathbf{z}) + \sum_{i,j} \frac{\partial^2 f(\mathbf{z})}{\partial z_i \partial z_j} B_{ij}(\mathbf{z})\right] p(\mathbf{z},t|\mathbf{y},t') \\ &+ \int\int d\mathbf{z} d\mathbf{x} f(\mathbf{z}) \left[W(\mathbf{z}|\mathbf{x},t)\, p(\mathbf{x},t|\mathbf{y},t') - W(\mathbf{x}|\mathbf{z},t)\, p(\mathbf{z},t|\mathbf{y},t')\right] + O(\varepsilon)\end{aligned}$$

This expression is very close to the final form of the DCKE in (2.1). There are a few extra steps – in particular, an integration by parts procedure pulls the A_i and B_i back into the spatial derivatives. However, the important point here is how the drift coefficients $A_i(\mathbf{z},t)$ and the diffusion coefficients $B_{ij}(\mathbf{z},t)$ refer to probabilistic steps in the range of the stochastic process. We could make this more precise by referring to the A_i coefficients as the expected value of steps along the coordinate axes in the state space of the stochastic process. The expected value of these steps from a current state are allowed to depend upon the current state as well as time attached to that current state. Note that the term "step" is misleading – what we're really referring to in A_i is the expected value of the derivative of x_i with respect to time. However, considering the first-order steps A_i is not enough. Note that even if the steps have an expected value of zero in all directions (i.e. $A_i = 0$ for all i), and therefore the process has constant mean, we could still describe the variance of the process along the i-th dimension (i.e. B_{ii}).

2.5.2 Jump Coefficients

Let us focus on the contribution of the third term in the DCKE to the dynamics of the transitional probability distribution.

$$\begin{aligned}\boxed{\partial_t\, p(z,t|y,t') =} & -\sum_i \frac{\partial}{\partial z_i}[A_i(z_i,t)p(z,t|y,t')] \\ & +\sum_{i,j}\frac{1}{2}\frac{\partial^2}{\partial z_i \partial x_j}[B_{ij}(z,t)p(z,t|y,t')] \\ & \boxed{+\int dx\,[W(z|x,t)\,p(x,t|y,t') - W(x|z,t)\,p(z,t|y,t')]}\end{aligned}$$

To understand the contribution of this third term, we must first analyze the meaning of the function W.

$$W(x|z,t) = \lim_{\Delta t\to 0}\frac{p(x,t+\Delta t|z,t)}{\Delta t}$$

where $|x-z| \geq \varepsilon$ and the convergence is uniform in x,z, and t

Because of the boundedness of the current state x away from the future state z, this function therefore gives something like a "jump derivative" – the probability of instantaneously moving from one location to a bounded away place. Remember that W is defined only when $|x-z| \geq \varepsilon$, so any transitions governed by W from the current state (x) to the future state (z) must happen by discontinuous jumps.

Now we consider the third component of the DCKE as a whole. The third component of the DCKE describes jumping in gain-loss form:

$$\int dx\,\underbrace{[W(z|x,t)\,p(x,t|y,t')}_{\text{z's gains}} - \underbrace{W(x|z,t)\,p(z,t|y,t')]}_{\text{z's losses}}$$

Thus, the transitional probability distribution is affected by the contribution of discontinuous jumps – the discontinuous jumps into state z minus the discontinuous jumps out of jumps out of state z.

Let us first consider the gains from the jumps. The term "z's gains" says that the probability of moving into future state z rises because of elements that are currently located at any state x (bounded away from z), but which currently has a "jump derivative" into state z. We integrate such elements over all possible intermediate states x which might be feeding into the focal state z. Similarly, let us now consider the losses from jumps The term "z's losses" says that the probability of moving into future state z falls because of elements which are already currently located at focal state z, but which currently have a "jump derivative" out of state z. Note that because our equation wants to express the probability of moving into some future state z at some future time t, we condition the probabilities on starting at some previous state y at some previous time t'.

Any contribution from these jump derivatives will produce sample paths which are discontinuous. By definition, a stochastic process with continuous sample paths must have

$$\lim_{\Delta t \to 0} \frac{\int_{|x-z| \geq \varepsilon} p(x, t+\Delta t|z,t)}{\Delta t} = 0.$$

Thus, if $W(x|z,t) = 0$, then the integrand will be zero for all possible values of x, and thus, the condition for continuity is met.

On the other hand, we might want to consider a system whose transitional probabilities evolve only from jumps. In other words, imagine that we have ($A_i = B_{ij} = 0$). Then the DCKE can be reduced to the "master equation":

$$\partial_t \, p(z,t|y,t') = \int dx \, [W(z|x,t) \, p(x,t|y,t') - W(x|z,t) \, p(z,t|y,t')]$$

3 The Wiener Process

3.1 ...As a Special Case of Continuous Markov Dynamics

The above reasoning brings us directly to one special case of the DCKE (2.1), which arises when we consider the DCKE in the special case where our state space has a single dimension, and where there are no jumps, A=0, and B=1. Then our equation describes the probabilistic dynamic evolution of a stochastic trajectory through state-space known as the Weiner Process.

$$\partial_t\, p(w,t|y',t') = \frac{1}{2}\partial_{ww}\, p(w,t|y',t') \tag{3.1}$$

Given some initial condition in the form $p(w,t_0) = \delta(w - w_0)$ – that is if we assume that our process originates at some certain location in state space – then the solution to the DCK equation , is a Gaussian:

$$p(w,t|w,t_0) = \frac{1}{\sqrt{2\pi(t-t_0)}} \exp\left(-\frac{(w-w_0)^2}{2(t-t_0)}\right) \tag{3.2}$$

In other words $W \sim N(w_0, t-t_0)$. In particular, the mean of our stochastic process at time t remains constant at the place of origin w_0, and the variance of our stochastic process at time t grows identically with the elapsed time $t - t_0$:

$$E\left[W(t)\right] = w_0$$

$$Var\left[W(t) - w_0\right] = t - t_0$$

Since we may pick any t_0 we want, we also know the distribution of our increment increments (the changes in our stochastic process over some discrete sampling interval):

$$\Delta W \sim N(0, \Delta t) \tag{3.3}$$

And in particular,

$$E\left[\Delta W(t)\right] = 0$$

$$Var\left[\Delta W(t)\right] = \Delta t$$

This particular stochastic process W moving through state space is also known as ”Brownian motion.”

3.2 ...As The Limit of a Scaled Random Walk

However, despite its exotic properties, the standard Wiener Process can be approximated in distribution on any finite time interval by a scaled random walk (Kloeden & Platen 1999). Such ”discretized Borwnian motion” serves many

purposes: numerical simulations, intuition building, and theoretical argument construction via limiting processes.

To illustrate the technique, we can subdivide the unit interval $[0,1]$ into N subintervals

$$0 = t_0 < t_1 < \ldots < t_k < \ldots t_N = 1$$

of equal length $\Delta t = 1/N$ and construct a stepwise continuous random walk $S_N(t)$ by taking independent, equally probable steps of length $\pm\sqrt{\Delta t}$ at the end of each subinterval. We construct two point random variables X_n taking values ± 1 with equal probability. Then we define:

$$S_N(t_n) = (X_1 + X_2 + \ldots + X_n)\sqrt{\Delta t}$$

with

$$S_N(t) = S_N(t_n)$$

on $t_n \leq t < t_{n+1}$ for $n = 0, 1, 2, \ldots, N-1$, and with $S_N(0) = 0$. Then the random walk has independent increments $I_1 = X_1\sqrt{\Delta t}, I_2 = X_2\sqrt{\Delta t}, I_3 = X_3\sqrt{\Delta t}, \ldots$,where the mean of these increments is given by $\mu_I = E[I] = 0$ and the variance of these increments is given by $\mathrm{Var}(I) = \mathrm{E}[(I-\mu_I)^2] = E[(I)^2] = \Delta t$. Note that although the process S_n does not have independent increments, the increments I_n themselves are independent for the given subintervals. Thus, since the variance operator distributes over the sum of independent random variables $Var(X_1 + X_2) = Var(X_1) + Var(X_2)$, we can compute $Var(S_N(t))$ by considering the process as the sum of $\frac{t}{\Delta t}$ many independent increments of variance Δt, and thus

$$Var(S_N(t)) = \underbrace{\frac{t}{\Delta t}}_{\text{number of increments}} \cdot \underbrace{\Delta t}_{\text{variance of an increment}} = t$$

for $0 \leq t \leq 1$. Now obviously (by elementary limit laws from differential calculus), the constant function $Var(S_N(t)) \to t$ as $N = \frac{1}{\Delta t} \to \infty$. By appealing to the Central Limit Theorem (see Kloeden & Platen 1999), it can be argued that $S_N(t)$ converges in distribution as $N \to \infty$ to a standard Weiner Process.

3.3 ...As a Set of Axioms

The above considerations can be considered motivation for constructing an axiomatic definition of the Wiener process.

Definition 1 (Standard Wiener Process)**.** *A* standard Wiener process *or* standard Brownian motion *is a stochastic process W(t) with sample paths on $t \in [0,\infty)$ which satisfy the following four conditions:*

1. *The sample paths start at zero: W(0) =0 (with probability 1).*
2. *The sample paths are continuous.*
3. *Their increments are centered Gaussians with linearly growing variance: For $0 \leq s < t \leq T$, the random variable given by the increment $W(t) - W(s)$ is normally distributed with mean 0 and variance $t - s$. One visual image that I find helpful is to envision a set of line segments emanating from a tethered down point, W(s), and spreading across many possible W(t)'s, and the distribution of the right endpoints of these line segments is Gaussian normal, with variance equal to the horizontal component of the line segment.*
4. *Their increments are independent: For $0 \leq s < t < u < v \leq T$, the increments $W(v) - W(u)$ and $W(t) - W(s)$ are independent.*

3.4 ...As a Particular Type of a Gaussian Process

One might ask how large the above class of stochastic processes actually is. However, it can be shown that Wiener process is a Gaussian process (Shalizi 2012). Using that characterization, we can see which kinds of processes satisfy the axiomatic conditions.

Definition 2 (Gaussian Process)**.** *A real-valued stochastic process is Gaussian when all its finite-dimensional distributions are multivariate Gaussian distributions.*

Proposition 3.1. *The Wiener process is a Gaussian Process.*

Proof Following Shalizi (2012), we pick any times $t_0, t_1, \ldots, t_k$. Then by definitional axioms (3) and (4), the collection of random variables describing increments to the Wiener Process, $W(t_1) - W(t_0), W(t_2) - W(t_1), \ldots, W(t_k) - W(t_{k-1})$, are independent Gaussian random variables. But if X,Y are independent Gaussian random variables, then $(X, X+Y)$ is a multivariate Gaussian random variable. Thus, the k-dimensional column vector $(W(t_1) - W(t_0), W(t_2) - W(t_0), W(t_3) - W(t_0), \ldots, W(t_k) - W(0))$ is a multivariate Gaussian random variable. We can simplify this statement by definitional axiom (1), and say that $(W(t_1), W(t_2), W(t_3), \ldots, W(t_k))$ is a multivariate Gaussian random variable. Thus, the random variable has a multivariate Gaussian distribution. But since $t_1, t_2, \ldots t_j$ were arbitrary, as was the number of times k, then all finite-dimensional distributions are multivariate Gaussians. So the Wiener process is a Gaussian process. □

Now a Gaussian process is determined by its mean over time $\mu(t)$ and its covariance function $\text{cov}(X(s), X(t))$ (just as the distribution of a univariate Gaussian random variable is determined by its scalar mean and covariance – see Kallenberg 2002). The axiomatic properties can easily be used (see Shalizi 2012) to show that $\mathbb{E}(W(t)) = 0$, and for any time $s < t$, then $\text{Cov}(W(s), W(t)) = s$

Thus, we can fully characterize the Wiener process as a Gaussian process with properties (a) mean $\mathbb{E}(W_t) = 0$ and (b) $\text{Cov}(W_s, W_t) = s$ whenever $s < t$.

4 The Ornstein-Uhlenbeck Process

4.1 Rejecting the Wiener Process as a model for momentary neural evidence

Recall that we wanted a way to describe the continuous evolution of a stochastic variable over time. This continuously evolving stochastic variable would describe the momentary representation of a stimulus property, serving as the input to a neural integrator responsible for making a behavioral decision. The Wiener Process described so far provides a good model in some respects, and a bad model in other respects.

4.1.1 We want a diffusion process

The Wiener process is a *diffusion process*. The properties of a "diffusion process" are desirable to model the neural representation of momentary evidence. That's because we want our model of this neural representation to change continuously, and we want to be able to describe how its means and variance evolve over time.

Definition 3 (Diffusion Process)**.** *A diffusion process is any Markov process with continuous paths defined by a transition probability function satisfying the Chapman-Kolmogorov equation. In particular, a stochastic process is a diffusion process if the following limits exist for all* $\varepsilon > 0, s \geq 0, x \in \mathbb{R}$*:*

1. $\lim_{t \downarrow s} \frac{1}{t-s} \int_{|y-x|>\varepsilon} p(s,x;t,y)\, dy = 0$
2. $\lim_{t \downarrow s} \frac{1}{t-s} \int_{|y-x|<\varepsilon} (y-x) p(s,x;t,y)\, dy = a(s,x)$
3. $\lim_{t \downarrow s} \frac{1}{t-s} \int_{|y-x|<\varepsilon} (y-x)^2 p(s,x;t,y)\, dy = b^2(s,x)$

Here are some explanations of what these conditions mean:

1. This condition prevents the process from having any instantaneous jumps. (Essentially, the limit says that the total probability of jumping from the current location outside some ε ball goes to zero as the two time points get closer and closer together, and it does so at a faster rate than the rate at which the two time points go to zero).
2. The quantity $a(s,x)$ is called the *drift* of the diffusion process. The drift $a(s,x)$ describes the instantaneous rate of change of the mean of the process, given that $X(s) = x$.
3. The quantity $b^2(s,x)$ is called the *diffusion coefficient* of the process. The diffusion coefficient $b^2(s,x)$ describes the instantaneous rate of change of the squared fluctuations of the process, given that $X(s) = x$.

A diffusion process is exactly what we would like to describe the continuous evolution of noisy evidence in continuous time. Indeed, a diffusion process is simply the limiting case of a random walk, which makes instantaneous jumps in discrete time (Smith & Ratcliff 2007; Bogacz et al. 2006).

4.1.2 However, we do not want a martingale.

However, the Wiener process is also a *martingale*. The properties of a "martingale" are not desirable for modeling the neural representation of momentary evidence. A martingale process constantly readjusts its future expected value based on its current value.

To define a martingale, we'll first need to define a "natural filtration." A natural filtration records all the information about the "past behavior" of the stochastic process (and only that information).

Definition 4 (Natural Filtration)**.** *Let (Ω, F, P) be a probability space. Let (S, Σ) be a measurable space. Let T be a set of real-valued times. Let $X : T \times \Omega \to S$ be a stochastic process. Then the natural filtration of F with respect to X at time t is defined to be*

$$\mathscr{F}_t = \sigma\left\{ X_s^{-1}(A) \middle| s \leq t, A \in \Sigma \right\},$$

i.e. the smallest sigma algebra on Ω that contains all pre-images of Σ-measurable subsets of S for times up to t.

Now we can use the notion of a natural filtration to define a martingale.

Definition 5 (Martingale)**.** *A stochastic process $\{X_t : 0 \leq t < \infty\}$ is a martingale with respect to the natural filtration $\mathscr{F}_t$ and probability measure P if:*

- $\mathbb{E}_P(X_t) < \infty$ *for all* $t \geq 0$
- $\mathbb{E}_P(X_{t+s}|\mathscr{F}_t) = X_t$ *for all* $t, s \geq 0$

Finally, we can prove the the Wiener process is a martingale.

Proposition 4.1. *The Wiener Process $W(t)$ is a martingale with respect to its natural filtration.*

Proof

$$\begin{aligned}
\mathbb{E}[W(t+h)|\mathscr{F}_t] &= \mathbb{E}[W(t+h)|W(t)] && \text{Apply the "Markov property" (axiom 4)} \\
&= \mathbb{E}[W(t+h) - W(t) + W(t)|W(t)] && \text{Add and subtract same thing} \\
&= \mathbb{E}[W(t+h) - W(t)|W(t)] + \mathbb{E}[W(t)|W(t)] && \text{Apply linearity of expectation} \\
&= \mathbb{E}[W(t+h) - W(t)|W(t)] + W(t) && \text{W(t) is fully determined by time t} \\
&= 0 + W(t) = W(t) && \text{Apply the "Gaussian increments" property (axiom 3)}
\end{aligned}$$

□

Is this a good model for a neural representation of momentary evidence about a stimulus? Absolutely not; Brownian motion is a poor way to do this, as the above argument shows. The expected value of the process at any future state constantly readjusts to its current position. Thus, if we consider the Wiener process as a "scaled random walk," we could say that it has no "destination." In contrast, we would certainly want a model for neural evidence about a stimulus to have a destination. The neural representation should be tracking some mean value corresponding to an achieved state of belief about the dot coherence. The model for neural evidence should be noisy, but noisy deviations should tend to snap back into a mean behavior. That is, $E[X_{t+s}|F_t]$ should not equal X_t, but some value between X_t and μ_t. If we write the model we'd like as a diffusion process, then it should have a drift coefficient $a(t,x)$ that is a monotonic function of the negative difference from the mean $\mu_t - x_t$, whereas the Wiener process has a drift coefficient of $a(t,x) = 0$.

4.2 The Ornstein-Uhlenbeck Process as a model for momentary neural evidence

Thus, if we want to model neural representation of momentary evidence as a stochastic process, we will not use the Wiener Process – but, instead, a process known as the Ornstein-Uhlenbeck Process (or OU Process).

The stochastic differential equation for an OU Process, as stated in the introduction, is:

$$dx_t = \theta(\mu - x_t)\, dt + \sigma\, dW_t$$

The OU Process is said to be "mean-reverting" because the process tends to drift over time towards its current mean μ_t. The OU Process is a good model for neural representations of momentary evidence because it is a diffusion process which is "mean-reverting," rather than a martingale.

To fully understand the "mean-reverting" property of the OU Process, let us refer to some general theory about stochastic differential equations of the form

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t \tag{4.1}$$

Proposition 4.2. *Consider a stochastic differential equation of the form* (4.1)*. Assume that a and b are continuous, satisfy a Lipschitz condition, have a linear growth bound, and that the set of initial conditions is measurable with finite expected value* $\mathbb{E}[X_{t_0}^2] < \infty$ *Then the solution* X_t *of* (4.1) *for any fixed initial value* X_{t_0} *is a diffusion process on* $[t_0, T]$ *with drift* $a(t,x)$ *and diffusion coefficient* $b(t,x)$*.*

Proof This is Kloeden and Platen (1998)'s Theorem 4.6.1. □

Thus, the OU process has a "drift coefficient" of $a(t,x) = \theta(\mu - x_t)$. As desired, its drift coefficient $a(t,x)$ is a monotonic function of the negative difference from the mean, $(\mu - x_t)$. In fact, the drift coefficient is exactly this difference from the mean, scaled by an intensity factor $\theta > 0$.

Now we can understand the OU Process by decomposing it into two terms. That is, we see the change in the process dx_t as being the sum of a (scaled) "white noise term" σdW along with an additional "mean-reverting term" $\theta(\mu - x_t)\,dt$ which reverts the stochastic process back towards some consistent "attracting" value μ rather than having it restabilize at the new current value x_t.

$$\underbrace{dx_t}_{\text{OU dynamics}} = \underbrace{\theta(\mu - x_t)\,dt}_{\text{"mean reverting" term}} + \underbrace{\sigma\,dW_t}_{\text{"white noise" term}}$$

Thus, the OU model provides a "mean-reverting" diffusion process – precisely what we wanted to model the noisy, continuous neural representation of stimulus evidence. And indeed, OU models have been extremely useful to the fields of psychology and cognitive sciences. Mathematical models of neural decision-making have used the OU process to successfully predict behavioral decision times and error rates as a function of accumulation rate and decision criteria (Ratcliff 1978, Smith 2000). In fact, the Nobel-Prize winning Hodgkin-Hukley model, which mathematically describes how action potentials are generated and propogated, has been give rise to a Ornstein-Uhlenbeck leaky integrate and fire model (by way of a two-dimensional "phenomenalization" of Hodgkin-Huxley known as the Morris-Lecar Model; Ditlevsen 2012).

5 Applying Ito Calculus to the OU SDE

5.1 Why We Would Want To Integrate the OU SDE

The Ornstein-Uhlenbeck stochastic differential equation in (1.1),

$$dX_t = \theta(\mu - X_t)\,dt + \sigma\,dW_t,$$

has a clear interpretation as a diffusion process due to Proposition 4.2, which tells us that general stochastic differential equations can be interpreted as diffusion processes as long as certain assumptions are met. And thus, if we were to simulate the sample trajectories described by this stochastic differential equation, then we could vary the parameters θ and σ, and thereby vary quantities with known interpretations.

However, there are certain properties of the sample paths that remain relatively inaccessible to us, so long as we are dealing with the differential equation (for dX_t) rather than the explicit solution (for X_t). What, for instance, is the covariance function $\text{cov}\,(X_s, X_t) = \mathbb{E}[(X_s - \mu)(X_t - \mu)]$? What is the variance function $\text{var}(X_t) = \mathbb{E}[(X_t - \mu)^2]$? What is the stationary variance $\lim_{t\to\infty} \mathbb{E}[(X_t - \mu)^2]$? What is the timescale τ of autocorrelation?

In order to understand the stationary variance, the covariance function, and the autocorrelation timescale (let alone to investigate the effects of tweaking these quantities), we need to have the explicit solution. This would be highly desirable, because as the SDE is currently written (in (1.1)), there are "interactions" between parameters. For example, in the current formulation of the OU SDE, (1.1), we could tweak the intensity θ of the "deterministic part" of the equation, but we'd be simultaneously affecting the variance and covariance functions, and thereby the stationary variance and the autocorrelation time-scale, and it is not completely clear how the parameters θ and σ are interacting to determine these.

5.2 Difficulty Integrating the OU SDE

Our first step in explicitly solving the OU SDE in (1.1) might be to integrate it, in which case we'd get:

$$X_t = X_0 + \int_0^t \theta(\mu - X_s)\,ds + \int_0^t \sigma dW_s, \quad 0 \le t \le T.$$

However we run into an immediate problem in interpreting the second integral expression, which has the form

$$\int_0^t dW_s,$$

of the Wiener Process.

Proposition 5.1. *With probability 1, the Wiener Process has sample paths which are nowhere differentiable.*

Proof Following Shalizi (2012), we assume by way of contradiction that W(t) is differentiable at t_0. Then

$$\lim_{t\to 0} \frac{W(t,E) - W(t_0,E)}{t - t_0}$$

must exist, for some set E of positive measure. Call its supposed value $W'(t,E)$. That is, for every $\varepsilon > 0$, there must be some δ such that $t_0 - \delta \le t \le t_0 + \delta$ implies

$$\left| \frac{W(t,E) - W(t_0,E)}{t - t_0} - W'(t_0,E) \right| \le \varepsilon$$

Without loss of generality, take $t > t_0$. Then by the independent increments property, $W(t,E) - W(t_0,E)$ is independent of $W(t_0,E)$, and has a Gaussian distribution with mean zero and variance $t - t_0$. Therefore the differential ratio is $\mathcal{N}(0, \frac{1}{t-t_0})$. The quantity inside the value sign is therefore Gaussian with distribution $\mathcal{N}(-W'(t_0), \frac{1}{t-t_0})$. The probability that the absolute value expression exceeds any ε is therefore always positive. In fact, that probability can be made arbitrarily large by taking t sufficiently close to t_0. Thus, with probability 1, there is no point of differentiability. □

5.3 Defining the Ito Stochastic Integral

So we need some other procedure for analyzing processes which are defined by integrating against Brownian motion. In particular, we want to come up with a method of interpreting an "Ito Stochastic Integral" as below

$$I_t(C) = \int_0^t C_s \, dW_s \quad t \in [0,T] \tag{5.1}$$

where C is a stochastic process and W is the Wiener process.

The simplest possible approach would be to define the stochastic integral as a Riemann-Stieltjes Integral.

Definition 6 (Riemann-Stieltjes Integral)**.** *Define a partition of an interval [0,T] by*

$$\tau_n : 0 = t_0 < t_1 < \ldots < t_{n-1} < t_n = T$$

and an intermediate partition σ_n of τ_n by

$$\sigma_n : t_{i-1} \leq y_i \leq t_i \text{ for } i = 1, \ldots, n$$

Let f and g be two real-valued functions on [0,1] and define

$$\Delta_i g = g(t_i) - g(t_{i-1}), \;\; 1, \ldots, n$$

Then i the limit

$$S = \lim_{n\to\infty} S_n = \lim_{n\to\infty} \sum_{i=1}^{n} f(y_i)\Delta_i g$$

exists as mesh$(\tau_n) \to 0$ and S is independent of the choice of the partitions τ_n and their intermediate partitions σ_n, then S is called the Riemann-Stieltjes integral *of f with respect to g on [0,T], and we write*

$$S = \int_0^T f(t) \, dg(t)$$

However, it can be shown that an expression of the form in definition (5.1) cannot in general be defined as a Riemann-Stieltjes integral. Thus, we need some other approach. The general procedure, as defined in Mikosch (1998) follows the general procedure for defining a Lesbesgue Integral. The basic idea is to first note that the stochastic integrals of so-called "simple processes" (stepwise constant functions) *can* be defined as Riemann-Stieltjes integrals, and then to define general stochastic integrals as the mean-square limit of the stochastic integrals of simple processes.

Definition 7 (Simple Stochastic Process)**.** *The stochastic process $C = (C_t, t \in [0,T])$ is said to be* simple *if it satisfies the following properties: There exists a partition*

$$\tau_n : 0 = t_0 < t_1 < \ldots < t_{n-1} < t_n = T$$

and a sequence $(Z_i, i = 1, \ldots, n)$ of random variables such that

$$C_t = Z_n \qquad \text{if } t = T$$
$$C_t = Z_i \qquad \text{if } t_{i-1} \leq t < t_i, \; i = 1, \ldots n$$

and also the sequence (Z_i) is adapted to $(\mathcal{F}_{t_{i-1}}, i = 1, \ldots, n)$, that is, Z_i is a function of Brownian motion up to time t_{i-1} and satisfies $\mathbb{E}[Z_i^2] < \infty$ for all i.

As stated in Mikosch (1998), if we integrating such "simple" stochastic processes against Brownian motion, we get a Reimann-Stieltjes Integral:

$$\int_0^T C_s \, dW_s = \sum_{i=1}^{n} C_{t_{i-1}}(B_{t_i} - B_{t_{i-1}}) = \sum_{i=1}^{n} Z_i \Delta_i B.$$

Now, we note that for any member C of a conveniently broad class of integrand processes, there exist simple processes $C^{(n)}$ which converge to it in a mean square sense.

Proposition 5.2. *Let C be adapted to Brownian motion on [0,T]; that is, let C_t be a function of $B_s, s \leq t$. Let also the integral $\int_0^T \mathbb{E}[C_s^2] \, ds$ be finite. Then there is a sequence $(C^{(n)})$ of simple processes such that*

$$\int_0^T \mathbb{E}\left[(C_s - C_s^{(n)})^2 \right] ds \to 0$$

Proof See Kloeden and Platen (1992), Lemma 3.2.1 □

As a result, we can show that the sequence $(I(C^{(n)})$ of Ito stochastic integrals converges in a mean square sense to a unique limiting stochastic process $I(c)$.

Proposition 5.3. *Let C be a process satisfying the assumptions in the previous proposition. Then there exists a stochastic process I(C) on [0,T] such that*

$$\mathbb{E}\left[\sup_{0 \leq t \leq T} (I_t(C) - I_t(C^{(n)})^2 \right] \to 0$$

Proof See Mikosch (1998), Appendix A4. □

Definition 8 (Ito Stochastic Integral)**.** *The mean square limit I(C) is called the* Ito stochastic integral *of C. It is denoted by*

$$I_t(C) = \int_0^t C_s \, dB_s, \quad t \in [0,T].$$

5.4 Using Ito Stochastic Integral to Solve the OU SDE

Although the definition of the Ito Stochastic Integral is unwieldy and doesn't leave much room for intuition, the definition has yielded a number of tools useful for computations. For instance, with the above definition of the Ito Stochastic Integral, we gain "Ito's Lemma," which is the stochastic calculus counterpart of the chain rule. We also gain an important property called the "isometry" property.

Proposition 5.4. *The Ito Stochastic Integral satisfies the "isometry" property:*

$$\mathbb{E}\left[\int_0^t C_s \, dB_s \right]^2 = \int_0^t \mathbb{E}[C_s^2] \quad t \in [0,T]$$

Proof See Mikosch (1998) pp.110 □

Using Ito's Lemma, we can solve the OU SDE given in the introduction to this document.

Proposition 5.5. *The Ornstein-Uhlenbeck SDE given in* (1.1) *is solved by*

$$X_t = \mu + e^{-\theta t} X_0 + \sigma e^{-\theta t} \int_0^t e^{\theta s} dB_s$$

where the latter integral is interpreted as an Ito stochastic integral.

Proof See Mikosch (1998) Example 3.2.5, which applies Ito's Lemma to solve the SDE. □

Of course, the solution to the SDE is still difficult to interpret or conceptualize, due to the unwieldy definition of the Ito Stochastic Integral. However, we can use the tools of Ito Calculus to analyze the solution further. For instance, if we apply the isometry property, then we can immediately see that when the initial condition $X_0 = 0$, then

$$\text{var}\,(X_t) = \frac{\sigma^2}{2\theta}(1 - e^{2-\theta t})$$

5.5 Re-Expressing The OU SDE In a Practically Useful Way

The definitional expression for the OU SDE has been given in (1.1) as:

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t$$

where the parameters were interpreted as:

- μ = mean of process
- θ = relative strength of deterministic part
- σ = relative strength of stochastic part
- x_0 = initial condition

As suggested by the plot in the introduction, two of these parameters are easily interpretable. The initial condition x_0 affects where the process starts. The mean μ is the value towards which the paths (stochastically) converge over time. Thus, μ is something like a stochastic attractor for this system.

In contrast, the parameters θ and σ do not have more precise interpretations. For example, the parameter σ has the strange units of space$/\sqrt{\text{time}}$. It is not immediately clear how these parameters relate to the process's qualitative characteristics: its covariance function, its stationary variance, or its correlation timescale.

However, with the use of Ito Calculus (the solution to the OU SDE given in the previous section, as well as the various tools of Ito Calculus, such as isometry and Ito's Lemma), we can reexpress the OU SDE in a way that its parameters have direct intepretations.

Stationary Variance: First,we note (see Gardiner 4.4.4) that the variance of the process is given by:

$$var(x_t) = \left(var(x_0) - \frac{\sigma^2}{2\theta}\right)e^{-2\theta t} + \frac{\sigma^2}{2\theta} \tag{5.1}$$

In particular, the stationary (long-term) variance of the process, which we'll call ν, is:

$$\nu := \lim_{t\to\infty} var(x_t) = \frac{\sigma^2}{2\theta}$$

Thus, we can rewrite the process in a way that makes the long-term variance ν explicit:

$$dx_t = \theta(\mu - x_t)dt + \sqrt{2\nu\theta}\, dW_t$$

Correlational Timescale: Second, we note (see Gardiner 3.8.4) that the covariance function is given by:

$$cov(x_t, x_s) = \frac{\sigma^2}{2\theta} e^{-\theta|t-s|} \tag{5.2}$$

Thus, $x(t)$ and $x(s)$ are only significantly correlated if

$$|t-s| \sim 1/\theta$$

So we can define a new variable $\tau := 1/\theta$ to represent the time-scale of the process - the time-scale at which the process has a significant autocorrelation. Thus, making the two substitutions described above, we obtain :

$$dx_t = \frac{(\mu - x_t)}{\tau} dt + \sqrt{\frac{2\nu}{\tau}}\, dW_t$$

where the parameters are interpreted as:

- μ = mean of process
- ν = variance of the process in its stationary state
- τ = time-scale at which the process has a significant autocorrelation

Now we can do things like contrast sample paths with varying degrees of ruggedness (correlational timescale) while keeping constant the process's mean and variance.

Note that it would have been difficult to do this without the tools of Ito Stochastic Calculus – as we required Ito Stochastic Calculus to determine what the variance and covariance functions were in the first place. Thus, if we had simply attempted to "add noise" to the differential equation, our perspicacity would have been severely limited – we could have tweaked the parameters for a very long time, perhaps indefinitely, without seeing the big picture.

Part III

The Adaptive Functionality of Robust Integrators

6 Robust Integration of Nonstationary Evidence

6.1 Motivation

Neurally plausible models of cognitive decision making are often centered upon the motion coherence task, in which the participant must make an ”upward” or ”downward” decision based upon incoming sensory evidence. The immediate sensory evidence is believed to be encoded by MT neurons, and the temporal integration of this evidence is believed to be done in the area LIP. In particular, the firing rate of various ”population codes” in the LIP reflect the current strength of evidence for a particular perceptual decision. Models of neurally plausible decision making often deploy stochastic differential equations to model the evolution of evidence for one decision over the other. The ”noise” in these differential equations come from many sources. In particular, (a) the MT neurons are stochastic spike generators; (b) the LIP neurons receive input from MT cells that are not selective for the appropriate decision; and (c) neural integrators would seem to be plagued, in theory, by a ”mistuning” problem. (If leak and self-excitation are not perfectly balanced, then the neural integrator will not be able to integrate evidence in a useful way).

In their paper on ”Neural Integrators for Decision Making,” Cain, Barreiro, Shadlen, and Shea-Brown (2011) explored the potential ability of robust integrators to solve the mistuning problem. A ”robust integrator” throws away a certain amount of the input stream (interpreted as units of standard deviations of the input, $\hat{R}$). A ”simple integrator” does not do this – it integrates everything in the input stream. A priori, robust integrators would seem to present a fundamental trade-off in performance: on the one hand, an integrator should be robust to mistuning (favoring large $\hat{R}$); on the other hand, an integrator would want to be sensitive to the relevant input stimulus (favoring small $\hat{R}$). Thus, robustness to mistuning would seem to come at the expense of sensitivity to input.

Cain et al. (2011) discovered that robustness (chopping out the unexceptional parts of the input stream) protects the integrator against feedback mistuning, and it does so with little to no cost! That is, when the integrator is allowed to randomly slip into phases of over-excitement or over-dampening (at the timescale of a single decision), a robust integrator outperforms a simple integrator (when the robustness is set to be at a particular level). However, surprisingly, in the control case, when the integrators are perfectly tuned, the robust integrator performs at about the same level as the simple integrator. Thus, the robustness mechanism appears to protect against circuit mistuning with little to no cost on performance.

In other words, there is a statistical interaction effect between type of integrator (robust vs. simple) and mistuning levels (perfectly tuned vs. mistuned) on performance. This interaction is particularly interesting, because it complicates any post hoc claim that the finding is obvious. (If robustness always won, then that would be because it’s ”throwing away the noise”; if the simple integrator was always superior, then that would be because it’s ”preserving all the information”). Somehow the robust integrator seems to be weeding out the noise due to mistuning, without sacrificing information about the signal.

What remains unknown, however, is whether the same robustness mechanism could have other advantages as well. An open question is systems neuroscience is how neural decision-making circuits deal with the problem of non-stationary inputs (Tsetsos et al. 2011). In particular, it is not clear how neural circuits ”ignore” the incoming noise during periods where there is no stimulus (Purcell et al. 2012). Delayed onset of stimuli seems like it would decrement the performance of a decision-maker, because it would increase the variance of the initial conditions, which as we saw in Equation (5.1), would increase the variance of the stochastic process providing input to the

integrator, and thereby decrement performance.

Thus, the simulations in this section aim to investigate whether robust integrators could help to protect decision-making mechanisms against the problem of non-stationary evidence – in particular, delayed stimulus onsets, and thereby delayed onset of ”meaningful” signal from direction-sensitive neurons in MT.

6.2 Methods

6.2.1 Firing rate of MT Neurons

We begin by modeling the pools of rightward vs. leftward selective sensory MT neurons. Then mean firing rate $\mu_{l,r}$ for the leftward (l) and rightward (r) pools are a linear function of dot coherence expressed as a percentage (Britten et al., 1993). In this paper, we assume the evidence favors the rightward alternative. We use the formula suggested by Nick, where μ is in Hertz and C is a dot coherence percentage, $0 \leq C \leq 100$.

$$\mu_{l,r} = r_0 + b_{l,r}(C) \tag{6.1}$$

where the parameters r_0, b_l and b_r are derived from firing rates observed across a range of coherences. If the evidence favors the right alternative, $b_r = .4$ and $b_l = -.2$; if the right alternative is favored, these values are exchanged.

6.2.2 Input to Integrator: The Summed Output of MT Neurons

Cain et al. (2011) simulated spike trains from weakly-correlated pools of neurons in MT, an area of the brain which codes for direction of motion in a perceptual stimulus. These spikes were convolved with an exponential filter, and then summed to create a continuous stochastic output. This output was well-approximated by an Ornstein-Uhlenbeck process of the form below. The output served in turn as the input to the neural integrators in LIP. The output for a given MT pool (l = leftward-selective MT pool, r=rightward-selective MT pool) is expressed as:

$$dI_{l,r} = \frac{\mu_{l,r}(C) - I_{l,r}}{\tau}\, dt + \sqrt{\frac{2\nu_{l,r}(C)}{\tau}}\, dW_t$$

with the mean $\mu_{l,r}(C)$ dictated by Equation (6.1), and with the parameters $\nu_{l,r}(C)$ and τ free parameters to be fitted to behavioral data.

6.2.3 Race to Bound Integrators

Following Tsetsos, Usher and McClelland (2011), we differentiate between three main classes of models for decision-making:

1. *Drift Diffusion Models* Single variable tracks the cumulative difference between momentary stimulus support for one hypothesis versus the other.
2. *Race to Bound Models* Multiple accumulators which race each other to a decision criterion. These accumulators are independent from each other besides dependencies in the stimulus stream.
3. *Attractor Models* Like the race model, but the accumulators now directly influence each other's state (they compete with each other by way of mutual inhibition). For example, see Usher and McClelland's (2001) Leaky Accumulator Model.

In this document, we move away from the Drift-Diffusion Models used in Cain et al. (2011), and instead we investigate Race to Bound models, because some neurophysiologists believe that they comport more directly with neural decision makers (Mike Shadlen, personal communication), and because they stand a better change of generalizing to the case of multiple-alternative choices (Nick Cain, personal communication).

In other words, the neural circuit has two separate accumulators – each accumulating evidence for a different decision – and the accumulator which first achieves full activation is the one which corresponds to the agent's decision.

6.2.4 Simple Integrator Circuit

Our simple integrator circuit equation is:

$$\tau_E \frac{dE_{l,r}}{dt} = -E_{l,r} + (1+\beta)E_{l,r} + \kappa I_{l,r}(t) \tag{6.2}$$

The three terms in this equation account for leak, feedback excitation, and sensory input (scaled by a weight κ), respectively. The parameter β is a random variable sampled on a trial by trial basis from a normal distribution with mean $\overline{\beta}$ and standard deviation σ_β. Whereas Cain et al. (2011) studied the effect of integrator mistuning on performance (e.g. the case where $\sigma_\beta \neq 0$, for the simulations in this document, we assume perfectly tuned integrators.

6.2.5 Robust Integrator Circuit

Our robust integrator circuit equation is:

$$\tau_E \frac{dE_{l,r}}{dt} = \begin{cases} 0 & : |\beta E_{l,r} + \kappa I_{l,r}| \leq R \\ \beta E_{l,r} + \kappa \Delta I_{l,r} & : \text{ otherwise} \end{cases} \tag{6.3}$$

The first line represents a series of potential wells as discussed in Cain et al (2011). In that paper, if the sum of the mistuned integrator feedback falls below the robustness limit R, the activity of the robust integrator remains fixed. In this document, we ask whether the same procedure can help to protect against non-stationary noise: because in our simulations, the integrators are not mistuned, the robust integrator simply discards input activity below a robustness limit. To interpret the robustness limit, it is convenient to normalize by the standard deviation of the input signal:

$$\hat{R} = \frac{R}{STD[I_m(t)]}$$

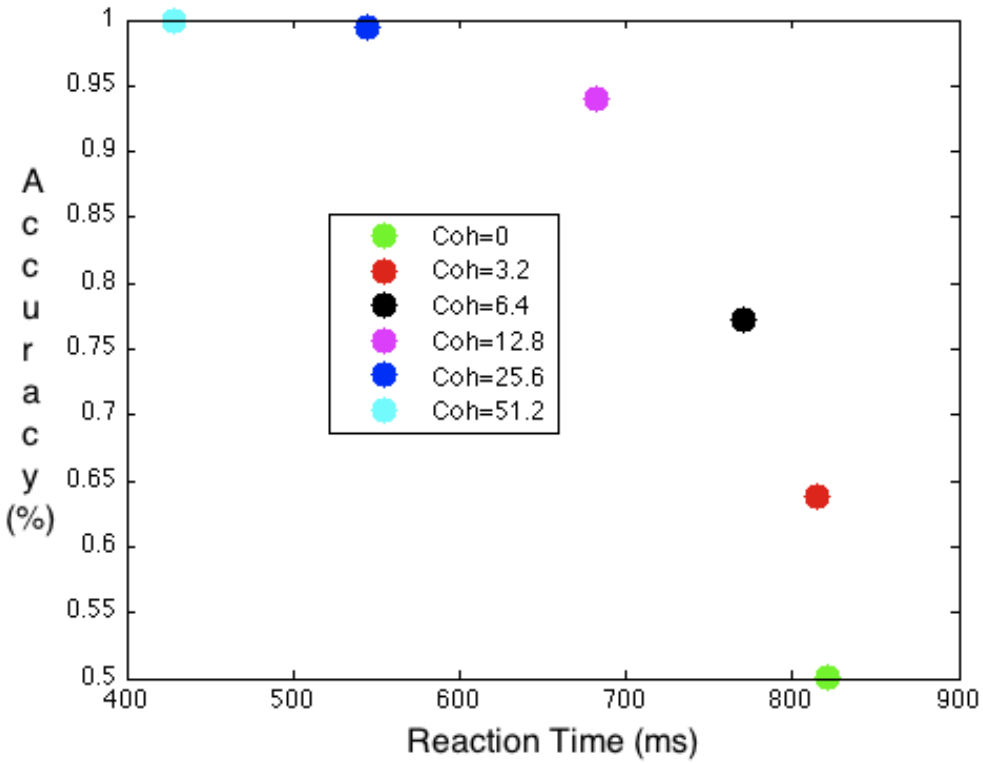
where the standard deviation is taken over all entries of the "mean input" signal $I_m = \frac{1}{2}(I_l + I_r)$.

6.3 Parameter Selection: Fitting Roitman's Data

Below is a table showing the parameter choices for our numerical simulations.
Many of the parameter choices we take to be fixed. The mean of the input signals was determined by the documented linear relationship between the stimulus property of dot coherence and the neural response in MT of mean firing rate (Britten et al. 1993). Since we are studying perfectly tuned integrators in this paper, we set $\sigma_\beta = 0$. In such a case, τ_E and κ can be combined into a single parameter, but that parameter simply shifts around the diffusion threshold, which we are already varying. Thus, for convenience, we set $\tau_E = \kappa = 1$. Following Cain et al. (2011), we set the robustness threshold to $\hat{R}$=.85. Finally, we set the simulation parameters in ways that do not interfere with the task at hand. In the end, we are left with a set of parameters where there are two free parameters, both of them within the input signal.

Parameter Choices	
Names	Value
Simulation Parameters	
Final Time Unit of Stored Input	tEnd=200
Step Size	dt=.01
Number of Simulations	ntrials= 5,000
Input Signal Parameters	
Dot Coherence	C=various
Mean of Input Signals	$\mu_1 = 20 - .2 * C$
	$\mu_2 = 20 + .4 * C$
Variance of Input Signals	$v = (\mu_1 + \mu_2)/FREEPARAMETER$
Correlation Timescale of Input Signals	τ_I = FREE PARAMETER
Integrator Parameters	
Integration Timescale	$\tau_E = 1$
Mistuning Parameter (Mean)	$\overline{\beta} = 0$
Mistuning Parameter (Std. Deviation)	$\sigma_\beta = 0$
Scaling Parameter (Internal Memory vs. Sensory Input)	$\kappa = 1$
Robustness Threshold	$\hat{R} = .85 *$ std. deviation of input signal

How can we set these parameters to "reasonable" values? The idea is to backwards-engineer them. In particular, we take a set of behavioral data recorded from monkeys performing in a dot coherence task (Roitman and Shadlen 2002). This data is plotted below, with the y-axis depicting accuracy (percent correct choices) as a function of response time (in milliseconds) The plotted points depend upon dot coherence in the stimulus ("Coh") – as coherence decreases, accuracy goes down, and reaction time goes up.



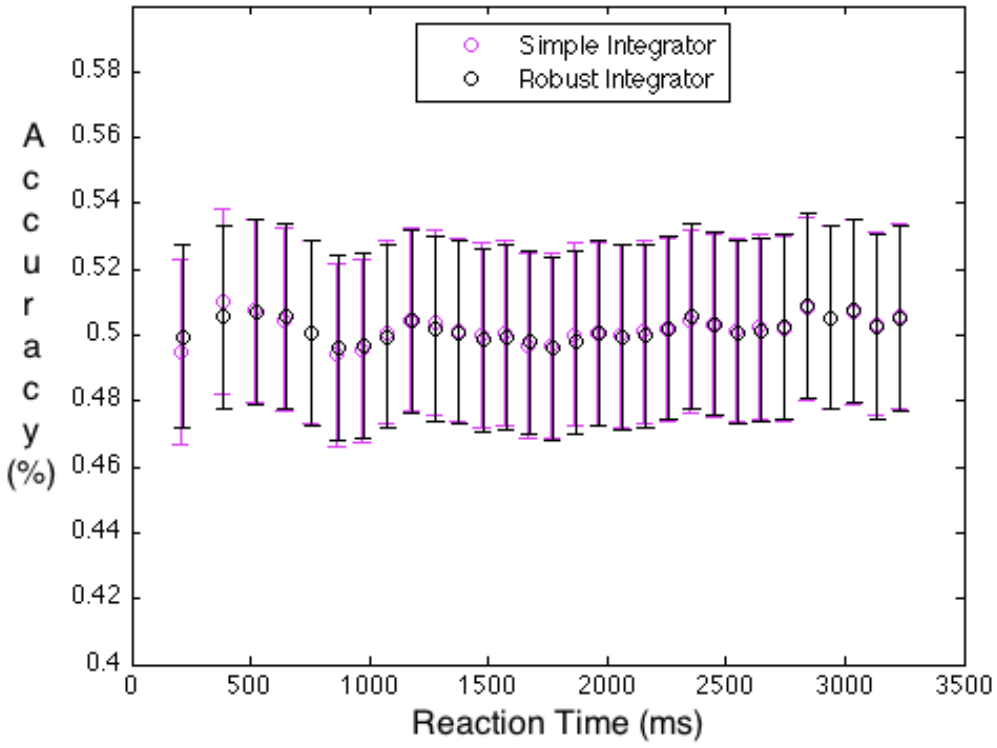
Using this data, we search the parameter space for ν and τ until we find a combination that fits the data fairly well for this set of coherence values. In other words, we search the parameter space until our model produces FC-RT (fixed choice/reaction time) curves which approximately go through these points. Based on this procedure of fitting the Roitman and Shadlen (2002) data, we end up choosing $\nu = (\mu_l + \mu_r)/4$ and $\tau = 4$. Thus, our final choice of parameters is as follows.

Parameter Choices	
Names	Value
Simulation Parameters	
Final Time Unit of Stored Input	tEnd=200
Step Size	dt=.01
Number of Simulations	ntrials= 5,000
Input Signal Parameters	
Dot Coherence	C=various
Mean of Input Signals	$\mu_1 = 20 - .2 * C$ $\mu_2 = 20 + .4 * C$
Variance of Input Signals	$\nu = (\mu_1 + \mu_2)/4$
Correlation Timescale of Input Signals	$\tau_I = 4$
Integrator Parameters	
Integration Timescale	$\tau_E = 1$
Mistuning Parameter (Mean)	$\overline{\beta} = 0$
Mistuning Parameter (Std. Deviation)	$\sigma_\beta = 0$
Scaling Parameter (Internal Memory vs. Sensory Input)	$\kappa = 1$
Robustness Threshold	$\hat{R} = .85 *$ std. deviation of input signal

6.4 Validity Checks

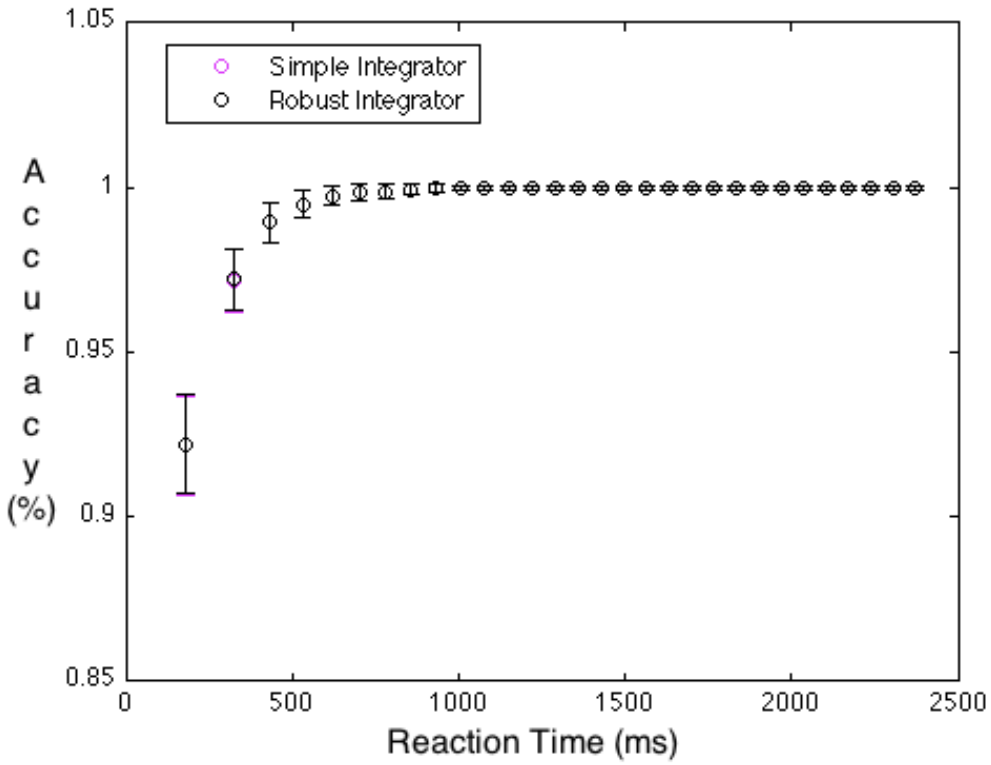
6.4.1 Bringing coherence down to zero

As the coherence level is brought down to zero, the performance of both integrators should approach 50/50. Below we show the graph of how the integrators performed at $C = 0$. This graph shows the results of averaging across 5,000 simulations. The plots show accuracy (percentage correct out of 5,000 trials) vs. reaction time (in milliseconds) to make a decision.



6.4.2 Bringing robustness threshold down to zero

As the robustness threshold is brought down to zero, the performance of the robust integrator should approach the performance of the simple integrator. Below we show the graph of how the integrators perform when $\hat{R} = 0$. This graph shows the results of averaging across 5,000 simulations at a coherence of $C = 25$. The plots show accuracy (percentage correct out of 5,000 trials) vs. reaction time (in milliseconds) to make a decision.



6.5 Experiment 1: Does Robust Integrator Protect Against Non-stationary (Delayed Onset) Input Noise?

Hypothesis:
Our primary hypothesis is that robust integrator will improve performance under the case of non-stationary input. In particular, we test the delayed onset of the stimulus. Here, we construct a delayed onset task where the two input streams have a mean of zero for the first quarter of the viewing period (0 ms $\leq t \leq 5,000$ ms), then they both turn on for the rest of the viewing period (5,000 ms $\leq t \leq 20,000$ ms).

Results:

The plots in Figure 3 depict performance of the robust and simple integrators on a typical trial. In particular, the plots depict idealized activation (in firing rate Hz; the term "idealized" is used because our integrator is non-reflective and can therefore briefly attain negative values) as a function of time. As the plots show, the robust integrator cuts out noisy changes in integrator state before the appearance of the stimulus at 5,000 ms.

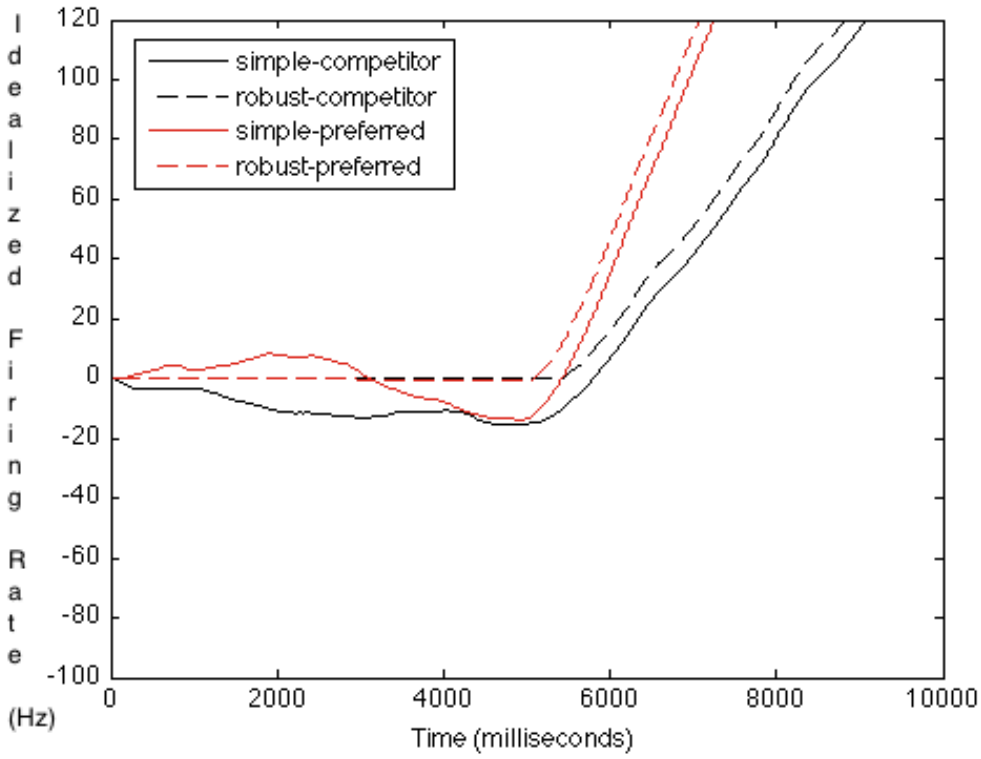


Figure 3: Single Trial Integrator Trajectories for Experiment 1

The plots in Figure 4 depict aggregate performance across all 5,000 simulated trials. These plots show accuracy (percentage correct out of 5,000 trials) vs. reaction time (in milliseconds) to characterize the performance of the integrators in making a decision. Since the black curve lies above the pink curve, we conclude that the robust integrator performs significantly better than the simple integrator in the current experiment. For a fixed reaction time, the robust integrator makes a more accurate decision. Likewise, for a fixed level of accuracy, the robust integrator makes a quicker decision. These results suggest that a robust integrator mechanism can help to improve decision-making in the presence of delayed stimulus onsets.

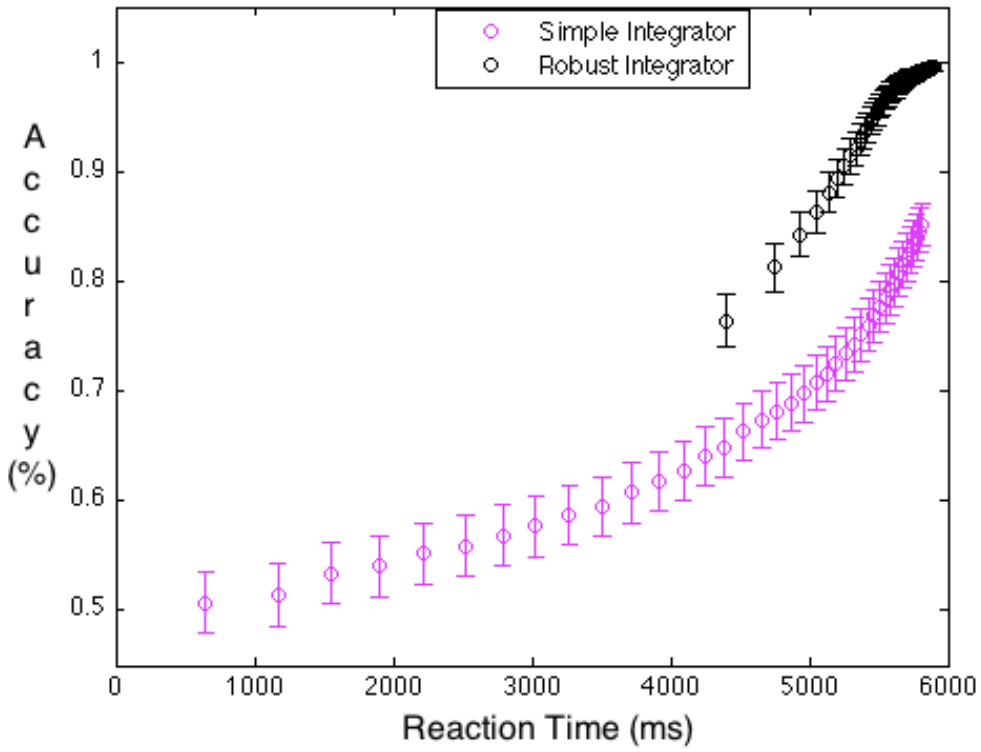


Figure 4: Accuracy-Speed Tradeoff Plot for Experiment 1

6.6 Experiment 2: How Do the Integrators Compare When Input Is Stationary ?

Hypothesis:
Perhaps the robust integrator is simply discarding information. In that case, we might expect that the robust integrator performs worse than the simple integrator when the stimulus onset is immediate. Thus, we construct the same decision-making task as in Experiment 1, except where the stimulus onset occurs immediately at the period of evidence accumulation.

Results:

The plots in Figure 5 depict performance of the robust and simple integrators on a typical trial. In particular, the plots depict idealized activation (in firing rate Hz; the term ”idealized” is used because our integrator is non-reflective and can therefore briefly attain negative values) as a function of time. As the plots show, the robust integrator and the simple integrator accumulate information in similar ways on a typical trial.

The plots in Figure 6 depict aggregate performance across all 5,000 simulated trials. These plots show accuracy (percentage correct out of 5,000 trials) vs. reaction time (in milliseconds) to characterize the performance of the integrators in making a decision. The black curve is essentially (and statistically) indistinguishable from the pink curve. This result suggests that that the robust integrator performs equally well as the simple integrator when the stimulus appears predictably at the beginning of the viewing period.

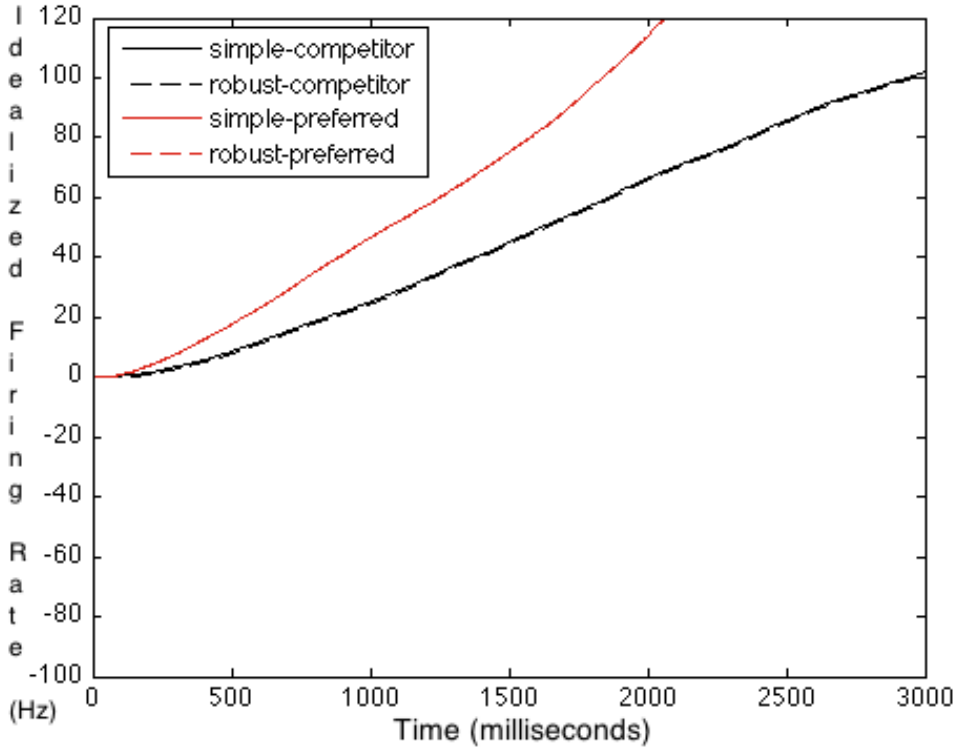


Figure 5: Single Trial Integrator Trajectories for Experiment 2

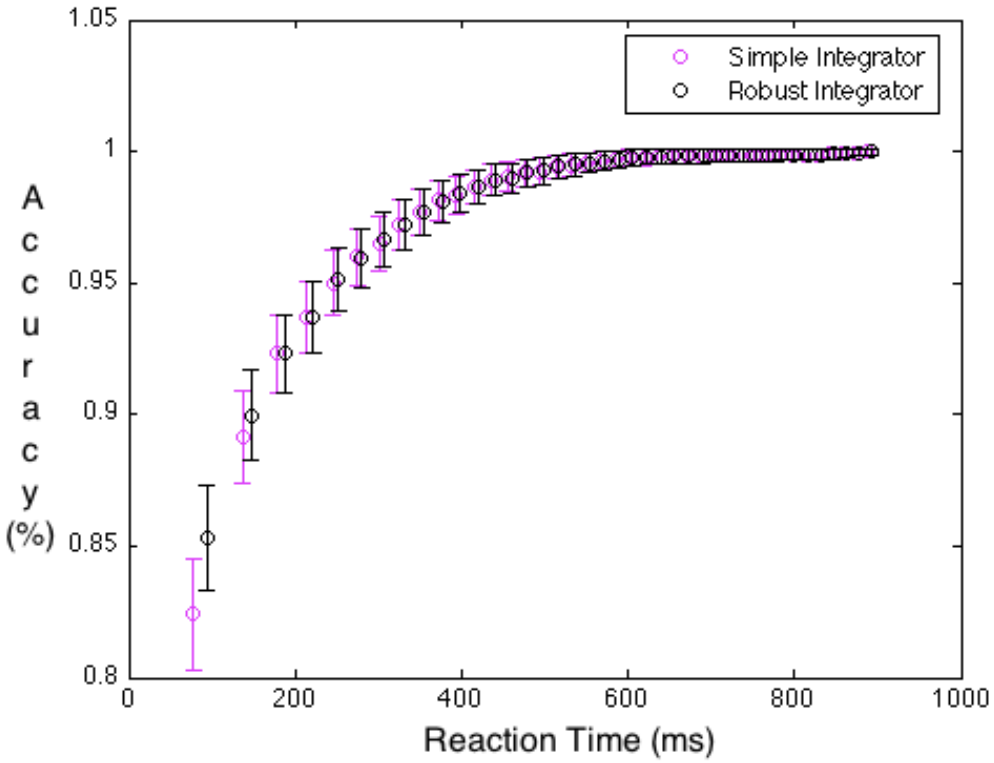


Figure 6: Accuracy-Speed Tradeoff Plot for Experiment 2

6.7 Conclusion and Future Directions

Taking the results of Experiments 1 and 2 in tandem, the results suggest that the robust integrator mechanism can help to improve decision-making in the presence of delayed stimulus onsets, apparently without a cost to decision-making in the presence of stationary inputs.

In conclusion, our simulations suggest that a robust integrators may have a dual purpose. Not only might they protect against integrator mistuning (as in Cain et al., 2011), they also seem to protect integrator circuits from noise that occurs due to non-stationary input streams. Future research should investigate these potential roles in tandem. Does the same model exhibit both functionalities? What are the boundary conditions? Do the results extend to multiple alternatives?

Part IV

Appendix

A Introduction to the Appendix

In the appendix, we include MATLAB files that were newly constructed by the author of the master's thesis for the purpose of testing hypothesis about robust integration of non-stationary input data.

B The Main Coding Loop

```
% Note: I've used a two integrator implementation
% Note: Here "1" means the competitor; "2" means preferred

clear all;
close all;
clc

ntrials=5000; %number of simulation trials (number of stochastic inputs)

% ------GENERATE INPUT SIGNAL ---------
C=25; % Dot Coherence (free to select)
mu1= 20-.2*C;
mu2= 20+.4*C; %mean input signal (determined by coherence)
nu=(mu1+mu2)/4; %variance input signal (fit to Roitman's data)
tau = 4; %timescale input signal (fit to Roitman's data)

tBegin=.01;
tEnd=200;
dt=.01;
t=tBegin:dt:tEnd;
N=length(t); % N gives the number of timesteps ( = number of columns)

%Step function (time on and time off)
h_on=1;
h_off=N;

%Writing the non-stationary means
mu1_vector(1:h_on)=0;
mu1_vector(h_on:h_off)=mu1;
mu1_vector(h_off+1:N)=0;

mu2_vector(1:h_on)=0;
mu2_vector(h_on:h_off)=mu2;
mu2_vector(h_off+1:N)=0;

%This function implements a non-stationary input stream
```

```
[I1]=NewBatchNonStationaryFunctionOU(mu1_vector,nu,tau,tBegin,tEnd,dt,
    ntrials);
[I2]=NewBatchNonStationaryFunctionOU(mu2_vector,nu,tau,tBegin,tEnd,dt,
    ntrials);

% ------SIMPLE INTEGRATOR ----------

sigmabeta= .1; % mistuning parameter
kappa=1; % parameter scaling sensory vs. internal

[E1]=NewBatchIntegrate(I1,sigmabeta,kappa,tau,tBegin,tEnd,dt,ntrials);
[E2]=NewBatchIntegrate(I2,sigmabeta,kappa,tau,tBegin,tEnd,dt,ntrials);


% ------ ROBUST INTEGRATOR --------
rhat= .85; % robustness parameter
avgstd = ( std(I1(:))+std(I2(:)) ) /2; %std deviation of a matrix done by
    matlab command "std2"
r= rhat * avgstd;

[R1]=NewBatchRobustIntegrate(I1,sigmabeta,kappa,tau,r,tBegin,tEnd,dt,ntrials
    );
[R2]=NewBatchRobustIntegrate(I2,sigmabeta,kappa,tau,r,tBegin,tEnd,dt,ntrials
    );


%% ------BEHAVIORAL PERFORMANCE --------
%% Note: Since integrator "2" does preferred direction and "1" does
    competitor,
%% exceeding the threshold in the positive direction means a correct choice
    .
%

maxRT = tEnd/dt;
maxthetaindex=80;  %threshold parameter
E_thetastep=2; %threshold parameter
R_thetastep=2; %threshold parameter

E_maxtheta=maxthetaindex*E_thetastep;
R_maxtheta=maxthetaindex*R_thetastep;
E1(:,end)=E_maxtheta+dt; % Force the final timestep to (artificially) cross
    threshold
E2(:,end)=E_maxtheta+dt; % This will force a RT on all processes.
R1(:,end)=R_maxtheta+dt; % If neither integrator made a decision, then later
     a coin flip will make it.
R2(:,end)=R_maxtheta+dt;

DATA=zeros(maxthetaindex,10);

for thetaindex=1:2:maxthetaindex;
```

```
display(thetaindex);
thetaE = 0+E_thetastep*thetaindex;
thetaR = 0+R_thetastep*thetaindex;

% Compute the RT's (first threshold crossing) in a vectorized way
% (i.e. must find the first instance of threshold crossing in each row)

finaltime=(1/dt).*tEnd; %Assign this time to trials where nothing
    selected

A1=E1>thetaE;
idx = find(A1); % A1 has rows = trials, cols = steps
[r c] = ind2sub(size(A1),idx); % Lists rows and columns where the
    process exceeds threshold
rt_E1 = accumarray(r,c,[],@min); % For each row (trial), finds the
    minimum column where threshold exceeded
rt_E1(rt_E1==0)=finaltime; % These trials timed-out. They will pick
    highest activation

A2=E2>thetaE;
idx = find(A2); % A2 has rows = trials, cols = steps
[r c] = ind2sub(size(A2),idx); % Lists rows and columns where the
    process exceeds threshold
rt_E2 = accumarray(r,c,[],@min); % For each row (trial), finds the
    minimum column where threshold exceeded
rt_E2(rt_E2==0)=finaltime; % These trials timed-out. They will pick
    highest activation

B1=R1>thetaR;
idx = find(B1); % B1 has rows = trials, cols = steps
[r c] = ind2sub(size(B1),idx); % Lists rows and columns where the
    process exceeds threshold
rt_R1 = accumarray(r,c,[],@min); % For each row (trial), finds the
    minimum column where threshold exceeded
rt_R1(rt_R1==0)=finaltime; % These trials timed-out. They will pick
    highest activation

B2=R2>thetaR;
idx = find(B2); % B2 has rows = trials, cols = steps
[r c] = ind2sub(size(B2),idx); % Lists rows and columns where the
    process exceeds threshold
rt_R2 = accumarray(r,c,[],@min); % For each row (trial), finds the
    minimum column where threshold exceeded
rt_R2(rt_R2==0)=finaltime; % These trials timed-out. They will pick
    highest activation

sign_E=sign(rt_E1-rt_E2); %Which integrator was fastest? Returns 1 (E2
    faster),-1 (E1 faster), or 0.
sign_R=sign(rt_R1-rt_R2);
```

```
    sign_E(sign_E==0)=-2*( rand(1, length(find(sign_E==0))) >.5)+1; %
        Eliminate zeros by Coin toss between -1 and 1.
    sign_R(sign_R==0) =-2*( rand(1, length(find(sign_R==0))) >.5)+1; % (note
        that I wrote a linear function to convert coin tosses (0,1) to signs
        (1,-1)

    choice_E= 1.5 + .5* sign_E; % convert to a choice: 1=competitor, 2=
        preferred
    choice_R= 1.5 + .5 *sign_R;  % (note: I wrote a linear function to
        convert signs (-1,1) to choices (1,2)

    rt_E=min(rt_E1,rt_E2);
    rt_R=min(rt_R1,rt_R2);


    [trash pci_E]=binofit(size(find(choice_E-1)),ntrials);
    [trash pci_R]=binofit(size(find(choice_R-1)),ntrials);

    lo_pci_E=pci_E(1,1); %get 95% confidence interval
    up_pci_E=pci_E(1,2);
    lo_pci_R=pci_R(1,1);
    up_pci_R=pci_R(1,2);


    format short G;
    %DATA(thetaindex,:)=[mean(rt_E),mean(choice_E-1),std(rt_E),mean(rt_R),
        mean(choice_R-1),std(rt_R)];
    DATA(thetaindex,:)=[mean(rt_E),std(rt_E),mean(choice_E-1),lo_pci_E,
        up_pci_E,mean(rt_R),std(rt_R),mean(choice_R-1),lo_pci_R,up_pci_R];

    % The "-1" converts mean choices (1 or 2) into percentages correct.
end

%% ----SAVING AND PLOTTING --------
%Data to Save
save experiment3data.mat DATA
parameters=[C,nu,tau,tEnd,dt,ntrials,h_on,h_off,sigmabeta,kappa,rhat,
    maxthetaindex,E_thetastep,R_thetastep];
parameterlabels='parameters=[C,nu,tau,tEnd,dt,ntrials,h_on,h_off,sigmabeta,
    kappa,rhat,maxthetaindex,E_thetastep,R_thetastep];';
save newestparameters.mat parameters parameterlabels

%Make Plot
figure;errorbar(DATA(:,1),DATA(:,3),DATA(:,5)-DATA(:,4),'mo');
hold on
errorbar(DATA(:,6),DATA(:,8),DATA(:,10)-DATA(:,9),'ko');
legend('Simple Integrator','Robust Integrator')
```

C The Non-Stationary Input Signal

```
%Simulating the Solution to the OU Process
%Using The Euler-Maruyama Method

function [x]=NewBatchNonStationaryFunctionOU(mu_vector,nu,tau,tbegin,tend,dt
    ,ntrials)

t=tbegin:dt:tend;
N=length(t);

IC=zeros(ntrials,1);
x=zeros(ntrials,N); % ntrials gives the number of trials (= number of rows)
x(:,1)=IC;

for i=2:length(x)
    x(:,i)=x(:,i-1)+ dt *(mu_vector(i)-x(:,i-1))/tau + sqrt(2*nu/tau)*sqrt(
        dt)*randn(ntrials,1);
end
```

D The Simple Integrator

```
function [E]=NewBatchIntegrate(I,sigmabeta,kappa,tau,tBegin,tEnd,dt,ntrials)

t=tBegin:dt:tEnd;
N=length(t); % N gives the number of timesteps ( = number of columns)
IC=zeros(ntrials,1);
beta=normrnd(0,sigmabeta,ntrials,1); % each trial gets its own normal random
    variable

E=zeros(ntrials,N); % ntrials gives the number of trials ( = number of rows)
E(:,1)=IC;
for i=2:length(E)
    E(:,i)=E(:,i-1)+ (1/tau)*(beta.*E(:,i-1) + kappa*I(:,i-1))*dt;
end
```

E The Robust Integrator

```
function [E]=NewBatchRobustIntegrate(I,sigmabeta,kappa,tau,robust,tBegin,
    tEnd,dt,ntrials)

t=tBegin:dt:tEnd;
N=length(t);
IC=zeros(ntrials,1);

%Beta - controls balance between decay and self-excitation
%Kappa - scales sensory input (relative to internal)

E=zeros(ntrials,N);
E(:,1)=IC;

Robust=ones(ntrials,1)*robust; %make vector of robustness thresholds

beta=normrnd(0,sigmabeta,ntrials,1); % each trial gets its own normal random
     variable

for i=2:length(E)
     toggler(:,i)=abs(beta.*E(:,i-1) + kappa*I(:,i-1)) > Robust; % outputs
         '1' if above robustness limit, '0' otherwise
     E(:,i)=E(:,i-1)+ (1/tau)* ( beta.*E(:,i-1) + (kappa*I(:,i-1).*toggler(:,
         i-1)) ) *dt; %pointwise multiplication
end
```

References

[1] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J.D. Cohen. The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological review*, 113(4):700, 2006.

[2] R. Bogacz and K. Gurney. The basal ganglia and cortex implement optimal decision making between alternative actions. *Neural computation*, 19(2):442–477, 2007.

[3] N. Cain, A. Barreiro, M. Shadlen, and E. Shea-Brown. The effect of robust integrator dynamics on decision-making performance. *BMC Neuroscience*, 11(Suppl 1):P97, 2010.

[4] N. Cain, A.K. Barreiro, M. Shadlen, and E. Shea-Brown. Neural integrators for decision making: A favorable tradeoff between robustness and sensitivity. *Arxiv preprint arXiv:1111.6573*, 2011.

[5] N. Cain and E. Shea-Brown. Computational models of decision making: integration, stability, and noise. *Current Opinion in Neurobiology*, 2012.

[6] S. Ditlevsen and P. Greenwood. The morris–lecar neuron model embeds a leaky integrate-and-fire model. *Journal of Mathematical Biology*, pages 1–21, 2012.

[7] C.W. Gardiner. *Stochastic methods*. Springer, 1985.

[8] J.I. Gold and M.N. Shadlen. The neural basis of decision making. *Annu. Rev. Neurosci.*, 30:535–574, 2007.

[9] D.J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, pages 525–546, 2001.

[10] O. Kallenberg. *Foundations of modern probability*. Springer Verlag, 2002.

[11] P.E. Kloden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. 1999.

[12] A. Longtin. Stochastic dynamical systems. *Scholarpedia*, 5(4):1619, 2010.

[13] T. Mikosch. *Elementary stochastic calculus with finance in view*, volume 6. World Scientific Pub Co Inc, 1998.

[14] B.A. Purcell, J.D. Schall, G.D. Logan, and T.J. Palmeri. From salience to saccades: multiple-alternative gated stochastic accumulator model of visual search. *The Journal of Neuroscience*, 32(10):3433–3446, 2012.

[15] R. Ratcliff. A theory of memory retrieval. *Psychological review*, 85(2):59, 1978.

[16] Cosma Shalizi. *Almost none of the theory of stochastic processes*. Online course notes, Math 754, Carnegie Mellon University, 2012.

[17] P.L. Smith. Stochastic dynamic models of response time and accuracy: A foundational primer. *Journal of Mathematical Psychology*, 44(3):408–463, 2000.

[18] P.L. Smith and R. Ratcliff. Psychology and neurobiology of simple decisions. *TRENDS in Neurosciences*, 27(3):161–168, 2004.

[19] J.M. Steele. *Stochastic calculus and financial applications*, volume 45. Springer Verlag, 2001.

[20] K. Tsetsos, M. Usher, and J.L. McClelland. Testing multi-alternative decision models with non-stationary evidence. *Frontiers in neuroscience*, 5, 2011.